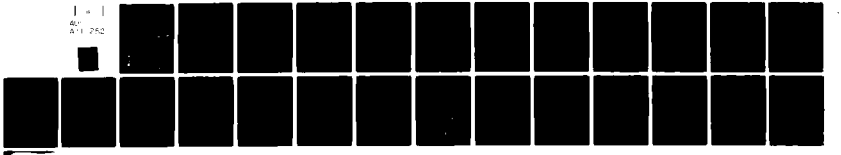
AD-A111 252 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2
A GPIB INTERFACE.(U)
JUN 81 A M BENSON
UNCLASSIFIED RAE-TM-RAD-NAV-168 DRIC-BR-79940 NL



END
DATE FILMED
3-82
DTIC

TECH. MEMO
RAD-NAV 168

TECH. MEMO
RAD-NAV 168

UNLIMITED

(2)

ROYAL AIRCRAFT ESTABLISHMENT

A GPIB INTERFACE

by

A. M. Benson

June 1981

34 p.

UNLIMITED

X

31445

ROYAL AIRCRAFT ESTABLISHMENT

Technical Memorandum Rad-Nav 168

Received for printing 19 June 1981

A GPIB INTERFACE

by

A. M. Benson

SUMMARY

This Memorandum shows the use of the Fairchild 96LS488 integrated circuit in the design of a GPIB interface for use with a Hewlett-Packard System 35A desktop computer. Circuits are described which will convert GPIB data into parallel BCD data and provide facility for hardware interrupts. The interface will both transmit and receive data.

This Memorandum expresses the opinions of the author and does not necessarily represent the official view of the Royal Aircraft Establishment

LIST OF CONTENTS

1 INTRODUCTION

When it is required to control, send data to, or receive data from a unit by computer it is necessary to provide an electronic interface to connect the two together. There are a large number of interface standards and one such standard is the IEEE Standard Digital Interface for Programmable Instrumentation[1] which describes the "General Purpose Interface Bus" (GPIB), of which more details are given in part 2. There is a list of abbreviations given after Table 3.

This Memorandum describes a general application of the Fairchild 96LS488 integrated circuit for a GPIB interface, but specific references will be made to the author's application. These should be viewed as an example of the way in which the interface may be used. The author's requirement was to utilise the GPIB in order to send digital information to a digital to analogue converter (DAC) remotely using a Hewlett-Packard System 35A desktop computer. The analogue output of the DAC was amplified by a control unit and used to drive a rate table as part of a test facility for ring laser gyroscopes[2], as shown in Fig 1. A standard interface card was attached to the computer and a similar interface was required for the DAC unit which was to receive the data. The requirement of the DAC unit was to be presented with four parallel BCD characters on 16 lines, consisting of sign and 3 numbers, preceded by an 'advance' pulse and followed by a 'trigger' pulse.

2 DESCRIPTION OF THE GPIB

The General Purpose Interface Bus is designed to provide byte serial, bit parallel communication between programmable instruments in a system in which at least one instrument can assume the role of controller, and all can either be talkers, listeners or both. A talker can transmit data on the bus, a listener can receive data from the bus and a talker/listener can both transmit and receive data. Each instrument has a unique address and a talker/listener may have separate talk and listen addresses. At any one time there may be any number of listeners but only one talker. In this application the Hewlett-Packard System 35A acts as the controller.

The bus consists of 16 wires, of which 8 are data and 8 management, and to this bus up to 15 instruments may be connected in either a star or daisy-chain configuration, using the structure shown in Fig 2.

Data is transmitted using a 3 wire handshake ($\overline{\text{DAV}}$, $\overline{\text{NRFD}}$ and $\overline{\text{NDAC}}$). A timing diagram for this handshake is given in Fig 3. $\overline{\text{DAV}}$ is generated by the talker and $\overline{\text{NRFD}}$ and $\overline{\text{NDAC}}$ are generated by the listener(s).

The other five management lines are as follows:

$\overline{\text{ATN}}$ generated by the controller. When $\overline{\text{ATN}}$ is forced low by the controller, all instruments on the bus should interpret information on the bus as interface commands or addresses rather than data or instrument codes.

$\overline{\text{REN}}$ when driven by the controller causes the instruments to transfer to remote control from front panel control.

$\overline{IFC}$ when this line is driven by the controller it initialises the system, ie all instruments on the bus are forced into a known quiescent state.

$\overline{EOI}$ used either by the controller to mark the end of a message or by the controller in a parallel poll sequence.

$\overline{SRQ}$ used by instruments to attract the attention of the controller which will then conduct a serial or parallel poll to identify which instrument requested service.

The meanings of the mnemonics for the 16 GPIB lines are included in the list of symbols and abbreviations. All bus drivers are either 3 state or open collector and the bus lines are active low, hence, to be compatible with active high TTL circuits they must be driven by inverting transceivers.

3 DESCRIPTION OF THE 96LS488 INTEGRATED CIRCUIT

A summary of the functions of the relevant pins of the 96LS488 is given in Appendix A. The 96LS488 is a 48 pin TTL LSI integrated circuit running from a single 5 V supply. It may have the same or different talk and listen addresses and may have secondary addresses. There are fourteen selectable modes of operation, which are listed in Table 1. It will respond to a serial poll or a parallel poll and has a built-in clock. All of the bus lines comply with IEEE specifications and so do not need separate bus drivers. All instrument signals are LSTTL compatible.

Data handshake from the interface to the instrument is performed by RXST and RXRDY. Data handshake from the instrument to the interface is performed by TXST and TXRDY. Status handshake from the instrument to the interface is performed by STST and STRDY.

$\overline{RSV}$ is taken low by the instrument, in order to cause the interface to issue a service request, and should not be released until the request is serviced.

Pins A1 to A5 select the address to which the interface will respond, A5 being the most significant bit. Pins M0-M3 select the mode of operation. The circuit may be reset by taking the $\overline{MR}$ pin low. If secondary addressing is required this is done using an external multiplexer and ASEL. The device clear output $\overline{CLR}$ issues a negative pulse when the interface receives a Device Clear command from the bus. The trigger output $\overline{TRIG}$ issues a negative pulse when the interface receives a Device Trigger command from the bus. The drive bus output $\overline{DRB}$ is taken low to enable the bus drivers, while the data/status output indicates whether the information on the bus is data or status. (Data low, status high). The external timing components for the on-chip clock use pins $\overline{CP}$ and XTAL.

A complete description of the IC is given in the Fairchild 96LS488 data sheet[3].

4 DESIGN REQUIREMENTS

The interface will perform three basic functions: listening, talking and interrupting. The design requirements for the instrument used as an illustration will now be set out with respect to these three functions.

4.1 As a listener

The interface will receive serial bytes of 8 bit parallel data from the GPIB and present then on 16 data lines, consisting of four BCD characters. There are two control

lines, the 'Advance' pulse, preceding the data, to prime the instrument and a 'Trigger' pulse, when the data lines are set, to trigger the instrument to accept the data. The number being sent to the instrument will be displayed on seven segment displays.

4.2 As a talker

The interface will serially handshake 8 bit parallel bytes of data from the instrument to the bus, to be received by the controller.

4.3 As a channel for hardware interrupt to the computer

The interface will interrupt the computer when required to do so by a pulse on one of up to 10 interrupt lines, and provide the computer with an information byte, stating which interrupt source originated the interrupt. The information byte will be an ASCII character 0-9 with parity 0 or 1. A definition of the ASCII codes is given in Table 3.

The interrupt lines are wired to sockets on the back of the interface unit and may be used to provide interrupt channels to the computer for other non-programmable instruments or any other source.

5 CIRCUIT DESCRIPTIONS

A block diagram of the interface unit described in section 4 is given in Fig 4.

5.1 Interface circuit (Fig 5)

The bus pins on the 96LS488 are connected straight to the bus. Because they are 3 state drivers they need no external bus drivers. The data lines however do need bus drivers as they are TTL. The drivers used are AM26S12A quad inverting bus transceivers with open collector outputs which are for this purpose terminated as shown. The data bus drivers are enabled by both $\overline{DRB}$, and $\overline{D}/S$ in the low state.

The timing components shown on pins $\overline{CP}$ and XTAL give a clock frequency of approx 5 MHz.

The interrupt pulse from the interrupt encoder board holds $\overline{RSV}$ low until an information byte has been requested. This will cause a pulse on the interrupt reset output which will reset the interrupt encode circuit. This can also be achieved by pressing the 'IF RESET' button. The 'LOCAL' push button will request the controller to return the instrument to front panel control.

5.2 Data encode circuit (Fig 6)

The output statement in the Hewlett-Packard System 35A program is:

OUTPUT SCS "[SIGN],[1st NUMBER],[2nd NUMBER], [3rd NUMBER]"

This causes the GPIB to execute six transmit cycles, transmitting in sequence [SIGN], [1st NUMBER], [2nd NUMBER], [3rd NUMBER], [CR], [LF] on the data lines. During each cycle the chip decodes the GPIB signals and takes RXST high when the output data lines are valid. The RXST pulse is used to increment counter IC3, and to trigger IC1 which after a preset delay, to allow storage of the data, pulses RXRDY to complete the handshake.

The storage of the data in the output latches (IC5 to IC8) is controlled by counter IC3 via the BCD/decimal decoder IC2. Only the least significant four bits are used, the most significant four bits being ignored.

Initially IC3 is in the '0' state and the '0' output of IC2 is set. The sequence is then:

'ADVANCE' and sign bit:

1 Interface generates RXST for 'sign'.

2 RXST 'AND'ed with '0' output of IC2 gives 'ADVANCE' pulse to DAC unit.

3 RXST causes IC3 to count up 1, so output of IC2 is now 1.

4 IC2 output 1 enables IC5, so 'sign' present on inputs is clocked into latch IC5 and appears on its outputs.

5 After preset delay IC1 generates RXRDY, and so enables the transfer of the next data byte on the GPIB.

First number:

6 Interface generates RXST for the 'first number'.

7 RXST causes IC3 to count up 1, so output of IC2 is now 2.

8 IC2 output 2 enables IC6 so 'first number' present on inputs is clocked into latch IC6 and appears on its outputs.

9 After preset delay IC1 generates RXRDY.

Second number:

10 Interface signals RXST for 'second number'.

11 RXST causes IC3 to count up 1, so output of IC2 is now 3.

12 IC2 output 3 enables IC7 so 'second number' present on inputs is clocked into latch IC7 and appears on its outputs.

13 After preset delay IC1 generates RXRDY.

Third number:

14 Interface generates RXST for 'third number'.

15 RXST causes IC3 to count up 1 so output of IC2 is now 4.

16 IC2 output 4 enables IC8 so 'third number' present on inputs is clocked into latch IC8 and appears on its outputs.

17 After preset delay IC1 generates RXRDY.

'TRIGGER':

18 Interface generates RXST for carriage return character.

19 RXST causes IC3 to count up 1, so output of IC2 is now 5.

20 IC2 output 5 causes 'TRIGGER' to go high.

21 After preset delay IC1 generates RXRDY.

Reset internal counter:

22 Interface generates RXST for line feed character.

23 RXST causes IC3 to count up 1, so output of IC2 is now 6, and 'TRIGGER' goes low.

24 IC2 output 6 resets IC3 and hence output of IC2 is now 0.

25 After preset delay IC1 generates RXRDY.

The first RXST pulse of the sequence triggers IC4, which after 0.1 second generates a reset pulse to the counter IC3. This ensures that if the wrong number of data bytes are transmitted due to a program error or stray pulses, leaving IC3 in a non valid state, it is returned to the '0' state, prior to the normal transmission of the next data block.

5.3 Interrupt encode circuit (Fig 8)

The interrupt lines are used as inputs to a decimal to BCD converter IC1. After a delay generated by C1 and R1 to allow the data to settle, the BCD output is stored in latch IC2 to be presented on the data lines when the data is inhibited. Each interrupt line is trapped by two schmitt trigger NAND gates, for noise immunity, and the outputs of these traps are taken to an AND gate whose output goes low if any trap changes state. For simplicty only three of these traps are shown in Fig 8, but there may be up to nine. This output is used to force $\overline{\text{RSV}}$ low and to disable the data lines and substitute the information byte.

The interrupt number appears on the four least significant bits, and the next three bits are set to 110 which converts the whole byte to an ASCII numeric value.

The parity link need only be included if the character set of the controller being used has parity 1. This is not the case with the Hewlett Packard System 35A.

The interrupt reset pulse from the interface circuit is generated once the information byte has been read by the controller and is initiated by the TXST line. Pressing 'IF RESET' also causes an interrupt reset. This reset pulse resets all the traps (there will normally only be one trap set, but there will be more if any other line has attempted to interrupt meanwhile) and clears the latch IC2. With the traps reset RSV now goes high and releases the interrupt request.

6 HOW TO USE THE INTERFACE

The pin connections for the GPIB connector are given in Table 2.

6.1 As a talker

The data output lines from the instrument are connected to the D1-D7 inputs to the interrupt encode circuit. Under normal circumstances this data will be presented to the interface circuit to be transmitted on the bus, and only when an interrupt is requested will the data be substituted by the information byte.

The timing for the instrument to transmit is achieved by the TXST and TXRDY lines.

6.2 As a listener

When using the unit as a listener the data encode circuit will only produce sensible output if the data sent on the bus is in the correct format. It is possible to change the required format by changing which outputs of IC2 are used to gate the latches and which one is used to reset IC3. The circuit as shown will require a command to be sent in the format as shown in section 5.2. If however the correct format is not used the counter will be reset after a preset delay and the circuit will be ready for a command in the correct format.

6.3 As a channel for hardware interrupts

An interrupt must consist of a negative pulse of duration at least one microsecond, to allow for gate settling times. A change of state from one to zero is sufficient. When an interrupt is so requested on one of the nine inputs the identity of the source as ASCII 0-9 will be substituted for the current data byte on D1-D8. For reference an ASCII character set is given in Table 3. On the Hewlett Packard System 35A the interrupt servicing routine should contain firstly a binary read statement which will obtain the value of the information byte before any other read statements are executed, otherwise it will be lost. If the information byte is not required it should still be read with a dummy statement.

7 CONCLUSION

This Memorandum has attempted to show how the 98LS488 IC can be used in an application where data needs to be transmitted and received by an instrument connected to the GPIB, and where other instruments need to interrupt the computer for service requests. It is not an attempt to give a full description of the GPIB, but an illustration of how it is possible to interface simple instruments to the GPIB, and extend the capabilities of the computer system. The example used should show potential users of the GPIB system the relative ease by which instruments may be interfaced to the GPIB and a computer.

Appendix

IMPORTANT CONNECTIONS TO THE 96LS488 IC

$\overline{CP}$ (10 MHz Clock) Used to clock internal-state flip-flops and is divided down internally to generate the data settling delay. All output changes are synchronous with the negative clock edge. The $\overline{CP}$ input can be driven by an external oscillator, or used in conjunction with XTAL output, as an RC or crystal oscillator.

XTAL (Crystal) Used to connect a crystal or external timing components for the on chip oscillator.

$\overline{MR}$ (Active low Master Reset) Initialises all internal latches and is completely asynchronous. After a reset all outputs to the GPIB are passive HIGH and $\overline{RQS}$ is in the high-Z state, $\overline{R}$/L, $\overline{TRIG}$, $\overline{CLR}$, $\overline{DRB}$, ASEL, $\overline{TAD}$ and $\overline{LAD}$ are HIGH. $\overline{D}$/S, RXST, STST, and TXST are LOW.

M0-M3 (Mode Control Inputs) Define 1 of 14 possible operating modes of the 96LS488. M0-M3 are HIGH-true inputs.

A1-A5 (Device Address Inputs) Define the instrument address and originate from switches, PC jumpers or software-loaded register. When different talk and listen addresses are required or when the secondary address feature is used, these inputs must be externally multiplexed, using the ASEL output to control the multiplexer. A1-A5 are HIGH-true inputs (H=1,L=0) and thus have the opposite polarity to the $\overline{DIO1}$-$\overline{DIO5}$ addresses.

ASEL (Address Select Output) Selects, through an external multiplexer, the Talk/Listen or Primary/Secondary address input, depending on the operating mode selected, LOW for TALK or primary address, HIGH for Listen or secondary address.

$\overline{LAD}$, $\overline{TAD}$ (Address Status Outputs) Indicate the Listen-Address or Talk-Address status respectively. They are also activated in the Talk-Only and Listen-Only modes. The outputs are active low to facilitate driving LED indicator lamps.

RXST (Receiver Strobe Output) Forms part of the handshake logic to pass data bytes to the instrument. When addressed to listen, the 96LS488 takes RXST HIGH when a valid data byte is on the bus and holds it high until the instrument signals (via the RXRDY input) that it has processed the byte. RXST may be inverted and connected to RXRDY, in which case the 96LS488 will receive data bytes from the bus at a data rate determined solely by the bus handshake.

TXST (Transmitter Strobe Output) Forms part of the handshake logic to pass data bytes from the instrument to the bus. When addressed to talk, the 96LS488 takes TXST HIGH to signal the instrument that the bus has accepted the data. TXST does not go low again until the instrument has acknowledged that the byte has been accepted (via the TXRDY input). TXST may be inverted and connected to TXRDY, in which case the 96LS488 will transmit data bytes at a data rate determined solely by the bus handshake.

STST (Status Strobe Output) Forms part of the handshake logic to pass a status byte from the instrument to the bus during a Serial Poll sequence. It operates in conjunction with the STRDY input in the same way as the TXST and TXRDY signals. STST may be inverted and connected to STRDY, in which case the status byte will be repeated as long as the 96LS488 is addressed to talk.

RXRDY (Receiver Ready Input) Forms part of the handshake logic controlling the passing of data from the bus to the instrument. RXRDY is driven HIGH when the instrument is ready to receive a data byte and LOW to acknowledge the receipt of a data byte.

TXRDY (Transmitter Ready Input) Forms part of the handshake logic controlling the passing of data from the instrument to the bus. When the 96LS488 is addressed to talk, TXRDY is driven HIGH when the instrument has a data byte to send and LOW to acknowledge that the byte has been accepted by the bus.

STRDY (Status Ready Input) Forms part of the handshake logic controlling the passing of a status byte to the bus during a Serial Poll sequence. It operates in a similar fashion to TXRDY.

$\overline{\text{RSV}}$ (Request Service Input) Is pulled LOW by the instrument to request service and initiate an $\overline{\text{SRQ}}$ interrupt to the controller. The interrupt will be cleared if $\overline{\text{RSV}}$ goes high before it is serviced, but once the $\overline{\text{SRQ}}$ has been serviced $\overline{\text{RSV}}$ must go HIGH before another $\overline{\text{SRQ}}$ can be started.

$\overline{\text{CLR}}$ (Clear Output) Issues a negative pulse when the 96LS488 receives a Device Clear (DC) command, or when it is addressed to listen and receives a Selected Device Clear.

$\overline{\text{TRIG}}$ (Trigger Output) Issues a negative pulse when the 96LS488 is addressed to listen and receives a Device Trigger (DT) command.

$\overline{\text{DRB}}$ (Drive Bus Output) Taken low to enable an external data bus driver when the 96LS488 is addressed to talk and is in the Talker Active State. $\overline{\text{DRB}}$ can also be used to tell the instrument logic to fetch the first byte.

$\overline{\text{RQS}}$ (Requested Service Output) A 48 mA 3-state output, enabled during a Serial Poll response and driven LOW if the 96LS488 initiated a $\overline{\text{SRQ}}$. $\overline{\text{RQS}}$ can be directly connected to $\overline{\text{DIO7}}$ in applications where it is the only status information to be sent.

$\overline{\text{D}}$/S (Data/Status Output) Valid during the Talk addressed state and indicates to the instrument logic whether the information to be sent via the bus is to be data or status (LOW for data, HIGH for status). A status byte is sent only in response to a Serial Poll, and in this case $\overline{\text{D}}$/S may be used to control a multiplexer to select data or status as the source to the bus drivers.

$\overline{R}$/L (Remote/Local) Output) Goes LOW when the controller puts the instrument in the remote mode via the $\overline{REN}$ command.

$\overline{RTL}$ (Return to Local Input) Taken LOW to request return of the instrument to local control. $\overline{RTL}$ will put $\overline{R}$/L HIGH unless the controller has put the 96LS488 into Local Lock-Out State.

IST (Instrument Status Input) Used by the Parallel Poll logic, IST is compared with the logic state defined by $\overline{DIO4}$ during the past PPE command. If IST is in the defined state, the 96LS488 will make an affirmative response to the next IDY message by making the assigned $\overline{DIO}$ line LOW. Note that IST is a HIGH-true input while $\overline{DIO4}$ is LOW-true.

Table 1

MODES OF OPERATION OF THE 96LS488

Mode inputs				Operating mode	Function
M0	M1	M2	M3		
L	L	L	L	Off line	The device cannot take part in any GPIB operations
L	L	L	H	TON (LOW speed)	The device goes directly to the talk addressed state and can source data to the bus
L	L	H	L	LON	The device goes directly to the listen addressed state and can receive data from the bus
L	L	H	H	TON (HIGH speed)	As for TON (LOW speed)
L	H	L	L	T (LOW speed)	Talker Only, single address mode
L	H	L	H	TE (LOW speed)	Talker Only, extended address mode
L	H	H	L	T (HIGH speed)	Talker Only, single address mode
L	H	H	H	TE (HIGH speed)	Talker Only, extended address mode
H	L	L	L	L	Listener Only, single address mode
H	L	L	H	LE	Listener Only, extended address mode
H	H	L	L	T/L (LOW speed)	Talker/Listener, dual address mode
H	H	L	H	TE/LE(LOW speed)	Talker/Listener, extended address mode
H	H	H	L	T/L (HIGH speed)	Talker/Listener, dual address mode
H	H	H	H	TE/LE (HIGH speed)	Talker/Listener, extended address mode

The LOW speed talker option is utilised when open collector data drivers are used. The HIGH speed option is selected when three state drivers are used.

Table 2

GPIB PINOUT, 24-WAY AMPHENOL

1	$\overline{\text{DIO1}}$	
2	$\overline{\text{DIO2}}$	
3	$\overline{\text{DIO3}}$	
4	$\overline{\text{DIO4}}$	
5	$\overline{\text{EOI}}$	
6	$\overline{\text{DAV}}$	
7	$\overline{\text{NRFD}}$	
8	$\overline{\text{NDAC}}$	
9	$\overline{\text{IFC}}$	
10	$\overline{\text{SRQ}}$	
11	$\overline{\text{ATN}}$	
12	SHIELD	
13	$\overline{\text{DIO5}}$	
14	$\overline{\text{DIO6}}$	
15	$\overline{\text{DIO7}}$	
16	$\overline{\text{DIO8}}$	
17	$\overline{\text{REN}}$	
18	DAV	Twisted pair return lines Grounded at interface circuit
19	NRFD	
20	NDAC	
21	IFC	
22	SRQ	
23	ATN	
24	GROUND	

Table 3

ASCII CHARACTER CODES

ASCII Char.	Equivalent forms Binary	Octal	Dec
NULL	00000000	000	0
SOH	00000001	001	1
STX	00000010	002	2
ETX	00000011	003	3
EOT	00000100	004	4
ENQ	00000101	005	5
ACK	00000110	006	6
BELL	00000111	007	7
BS	00001000	010	8
HT	00001001	011	9
LF	00001010	012	10
V_{TAB}	00001011	013	11
FF	00001100	014	12
CR	00001101	015	13
SO	00001110	016	14
SI	00001111	017	15
DLE	00010000	020	16
DC_1	00010001	021	17
DC_2	00010010	022	18
DC_3	00010011	023	19
DC_4	00010100	024	20
NAK	00010101	025	21
SYNC	00010110	026	22
ETB	00010111	027	23
CAN	00011000	030	24
EM	00011001	031	25
SUB	00011010	032	26
ESC	00011011	033	27
FS	00011100	034	28
GS	00011101	035	29
RS	00011110	036	30
US	00011111	037	31

ASCII Char.	Equivalent forms Binary	Octal	Dec
space	00100000	040	32
!	00100001	041	33
"	00100010	042	34
#	00100011	043	35
$	00100100	044	36
%	00100101	045	37
&	00100110	046	38
'	00100111	047	39
(	00101000	050	40
)	00101001	051	41
*	00101010	052	42
+	00101011	053	43
,	00101100	054	44
-	00101101	055	45
.	00101110	056	46
/	00101111	057	47
ø	00110000	060	48
1	00110001	061	49
2	00110010	062	50
3	00110011	063	51
4	00110100	064	52
5	00110101	065	53
6	00110110	066	54
7	00110111	067	55
8	00111000	070	56
9	00111001	071	57
:	00111010	072	58
;	00111011	073	59
<	00111100	074	60
=	00111101	075	61
>	00111110	076	62
?	00111111	077	63

ASCII Char.	Equivalent forms Binary	Octal	Dec
@	01000000	100	64
A	01000001	101	65
B	01000010	102	66
C	01000011	103	67
D	01000100	104	68
E	01000101	105	69
F	01000110	106	70
G	01000111	107	71
H	01001000	110	72
I	01001001	111	73
J	01001010	112	74
K	01001011	113	75
L	01001100	114	76
M	01001101	115	77
N	01001110	116	78
O	01001111	117	79
P	01010000	120	80
Q	01010001	121	81
R	01010010	122	82
S	01010011	123	83
T	01010100	124	84
U	01010101	125	85
V	01010110	126	86
W	01010111	127	87
X	01011000	130	88
Y	01011001	131	89
Z	01011010	132	90
[	01011011	133	91
\	01011100	134	92
]	01011101	135	93
^	01011110	136	94
_	01011111	137	95

ASCII Char.	Equivalent forms Binary	Octal	Dec
`	01100000	140	96
a	01100001	141	97
b	01100010	142	98
c	01100011	143	99
d	01100100	144	100
e	01100101	145	101
f	01100110	146	102
g	01100111	147	103
h	01101000	150	104
i	01101001	151	105
j	01101010	152	106
k	01101011	153	107
l	01101100	154	108
m	01101101	155	109
n	01101110	156	110
o	01101111	157	111
p	01110000	160	112
q	01110001	161	113
r	01110010	162	114
s	01110011	163	115
t	01110100	164	116
u	01110101	165	117
v	01110110	166	118
w	01110111	167	119
x	01111000	170	120
y	01111001	171	121
z	01111010	172	122
{	01111011	173	123
|	01111100	174	124
}	01111101	175	125
~	01111110	176	126
DEL	01111111	177	127

LIST OF SYMBOLS/ABBREVIATIONS

ASCII	American standard code for information interchange
BCD	binary coded decimal
DAC	digital to analogue converter
GPIB	general purpose interface bus
IC	integrated circuit
LSI	large scale integration
LSTTL	low power Schottky TTL
TTL	transistor transistor logic

GPIB lines

$\overline{\text{ATN}}$	attention
$\overline{\text{DAV}}$	data valid
$\overline{\text{EOI}}$	end or identify
$\overline{\text{IFC}}$	interface clear
$\overline{\text{NDAC}}$	not data accepted
$\overline{\text{NRFD}}$	not ready for data
$\overline{\text{REN}}$	remote enable
$\overline{\text{SRQ}}$	service request
$\overline{\text{DIO1-8}}$	data input/output lines

REFERENCES

No.	Author	*Title, etc*
1	IEEE	IEEE Standard interface for programmable instrumentation. IEEE Std 488 (1978)
2	A.J. Lindop	Laser gyro test equipment. *Rad/Nav Working Paper* RNI-WP124
3	Fairchild	Data sheet for 96LS488

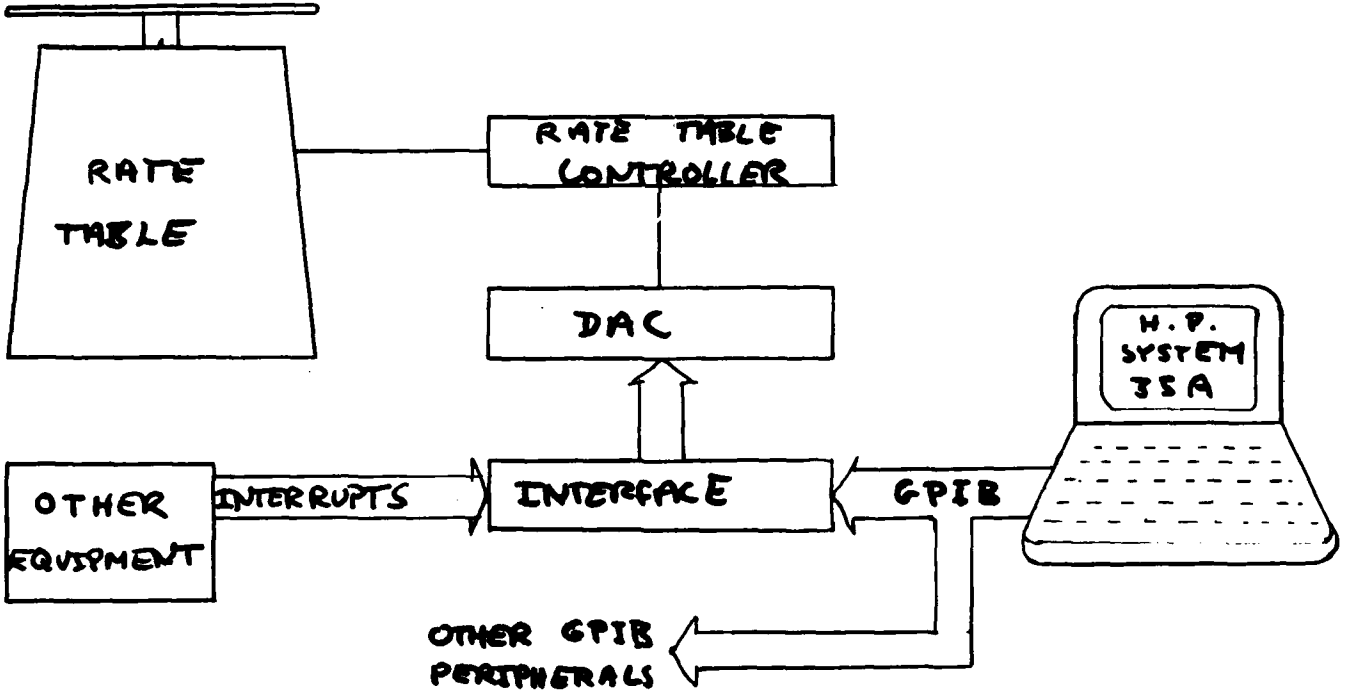

RATE
TABLE
RATE TABLE
CONTROLLER
DAC
H.P.
SYSTEM
35A
OTHER
EQUIPMENT
INTERRUPTS
INTERFACE
GPIB
OTHER GPIB
PERIPHERALS

Fig 1

Fig 1

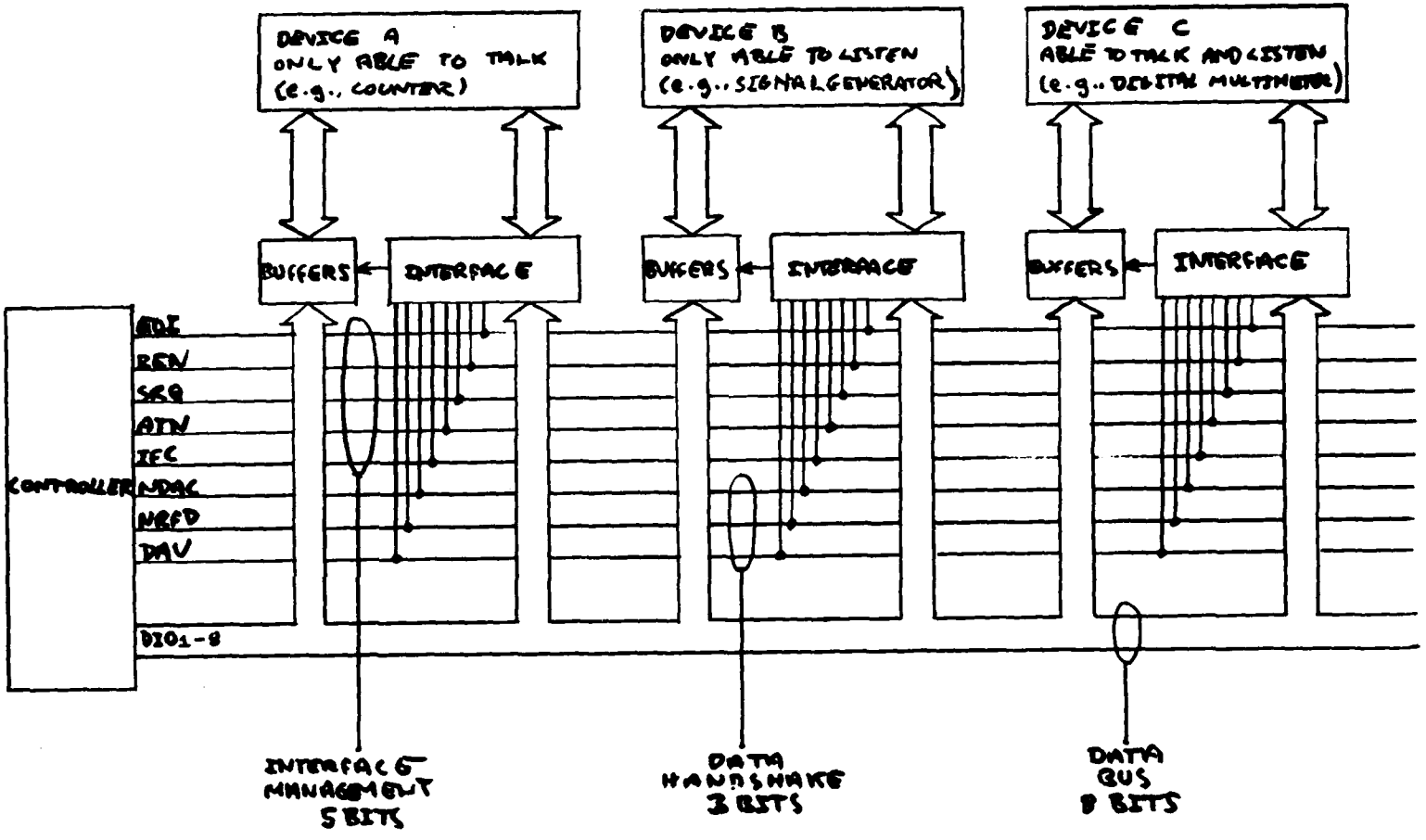


Fig 2 GPIB structure

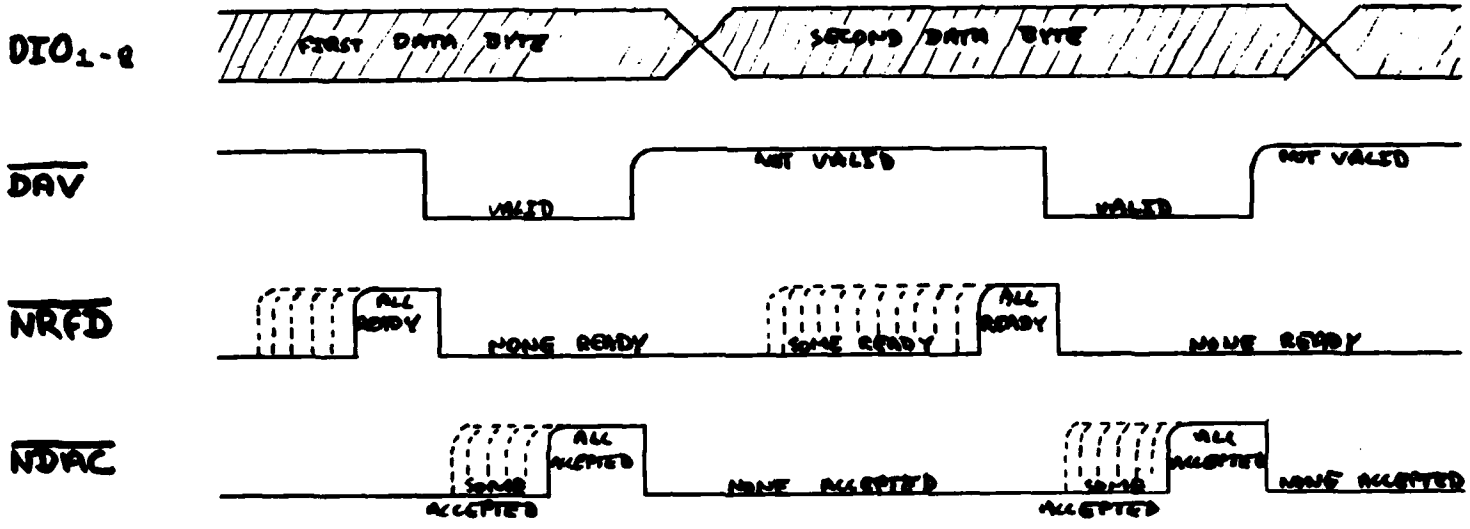


Fig 3 Data handshake timing sequence, one talker and multiple listeners

Fig 3

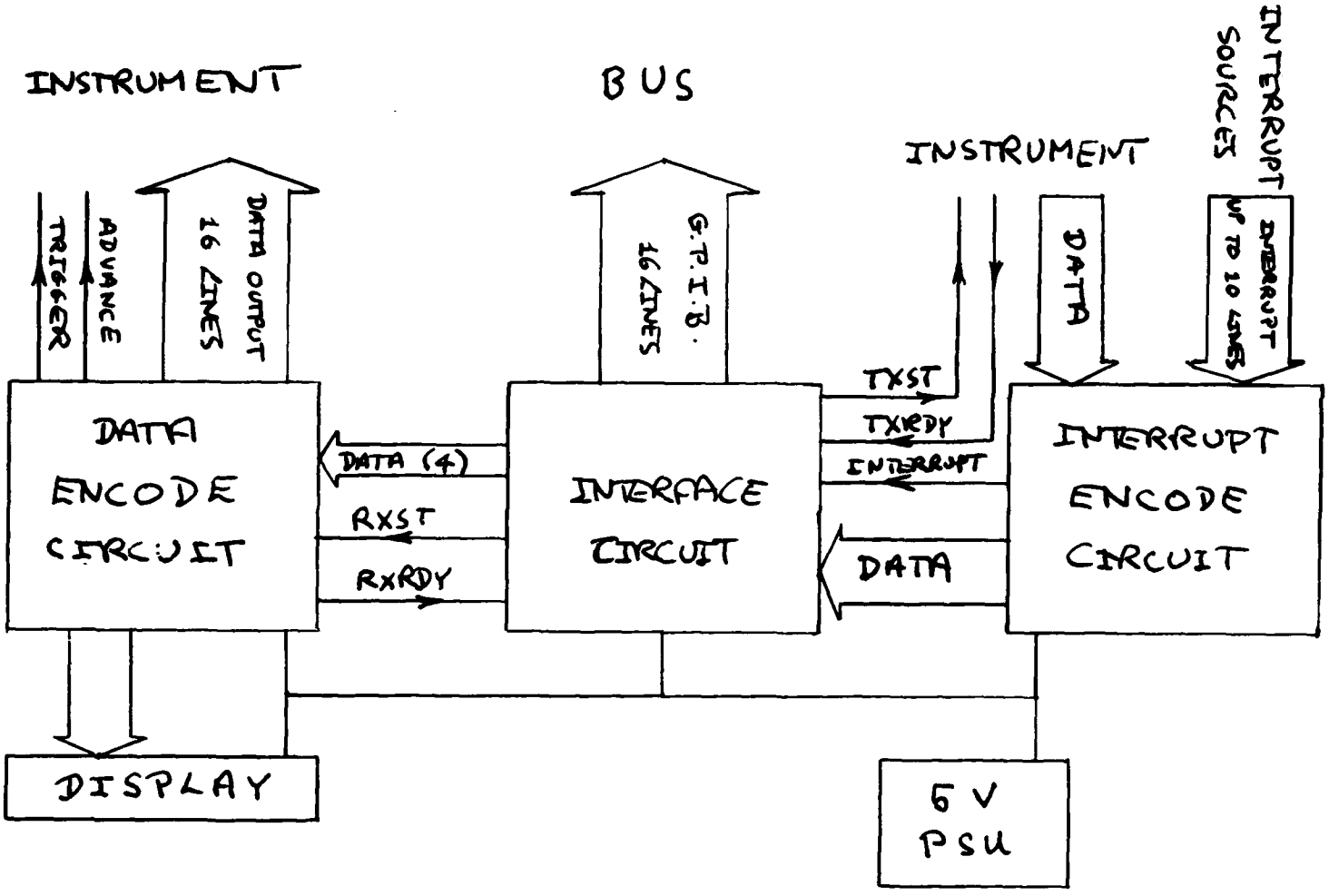


Fig 4 Block diagram of interface

Fig 4

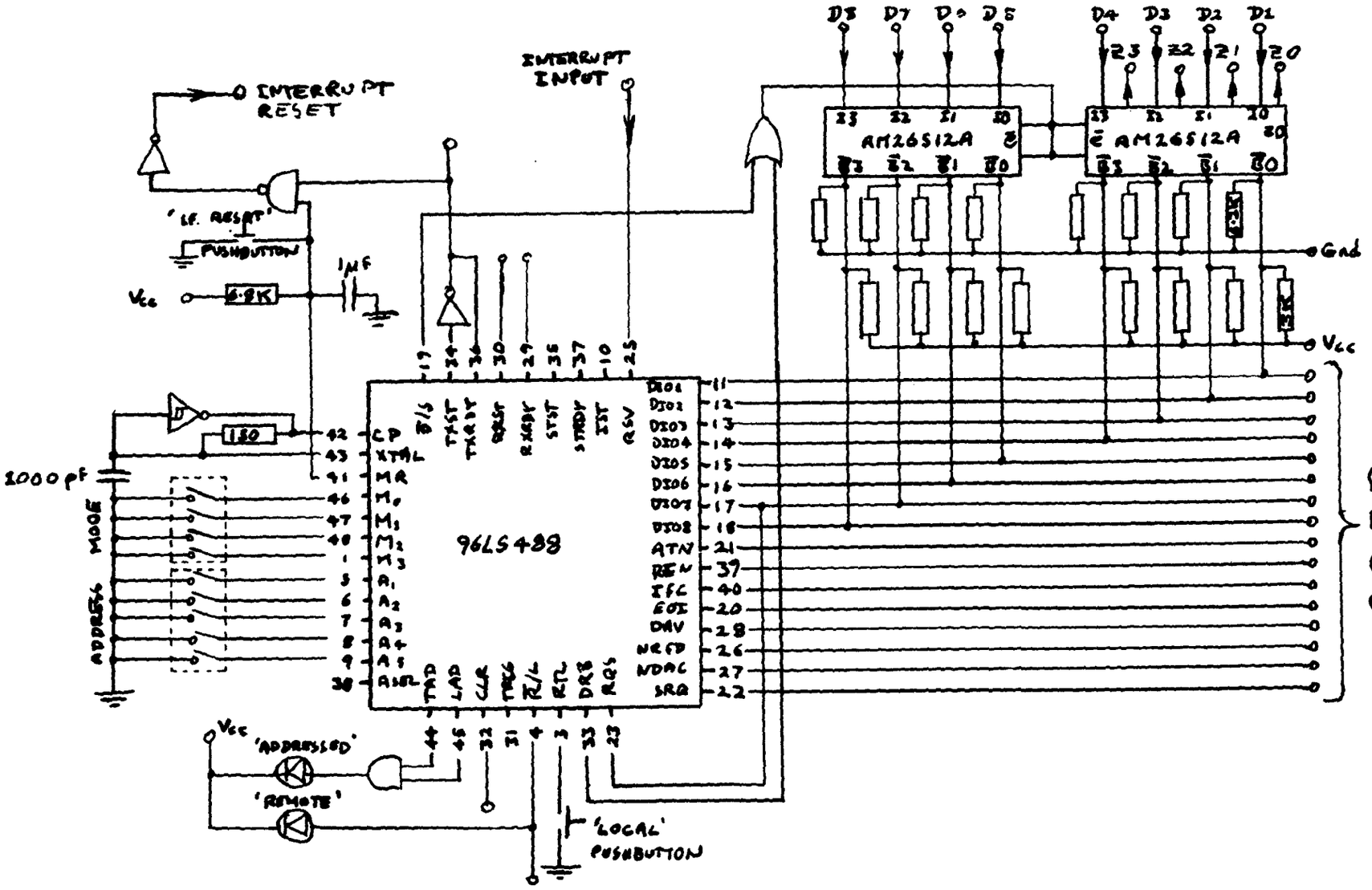


Fig 5 Interface circuit

Fig 5

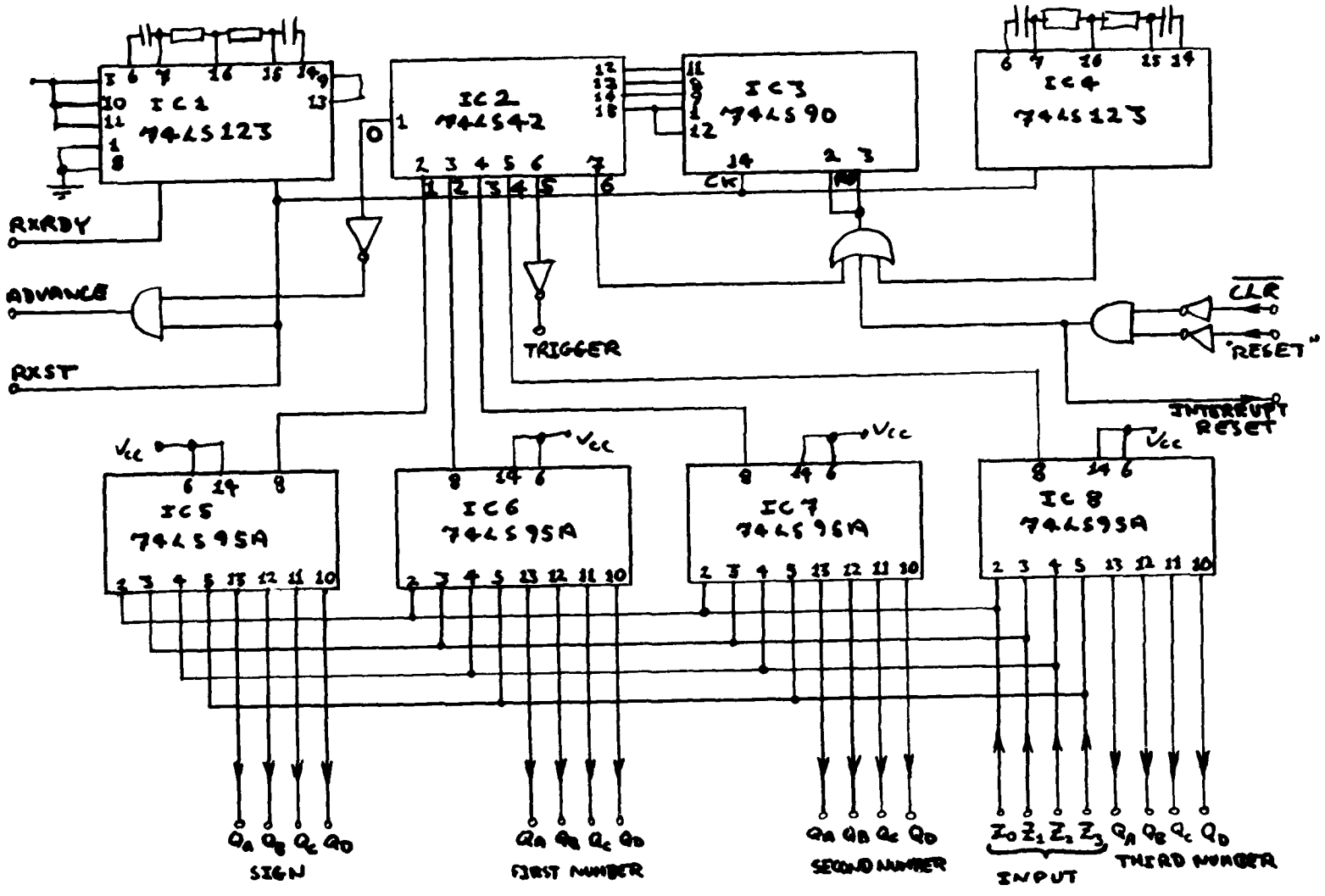


Fig 6 Data encode circuit

Fig 6

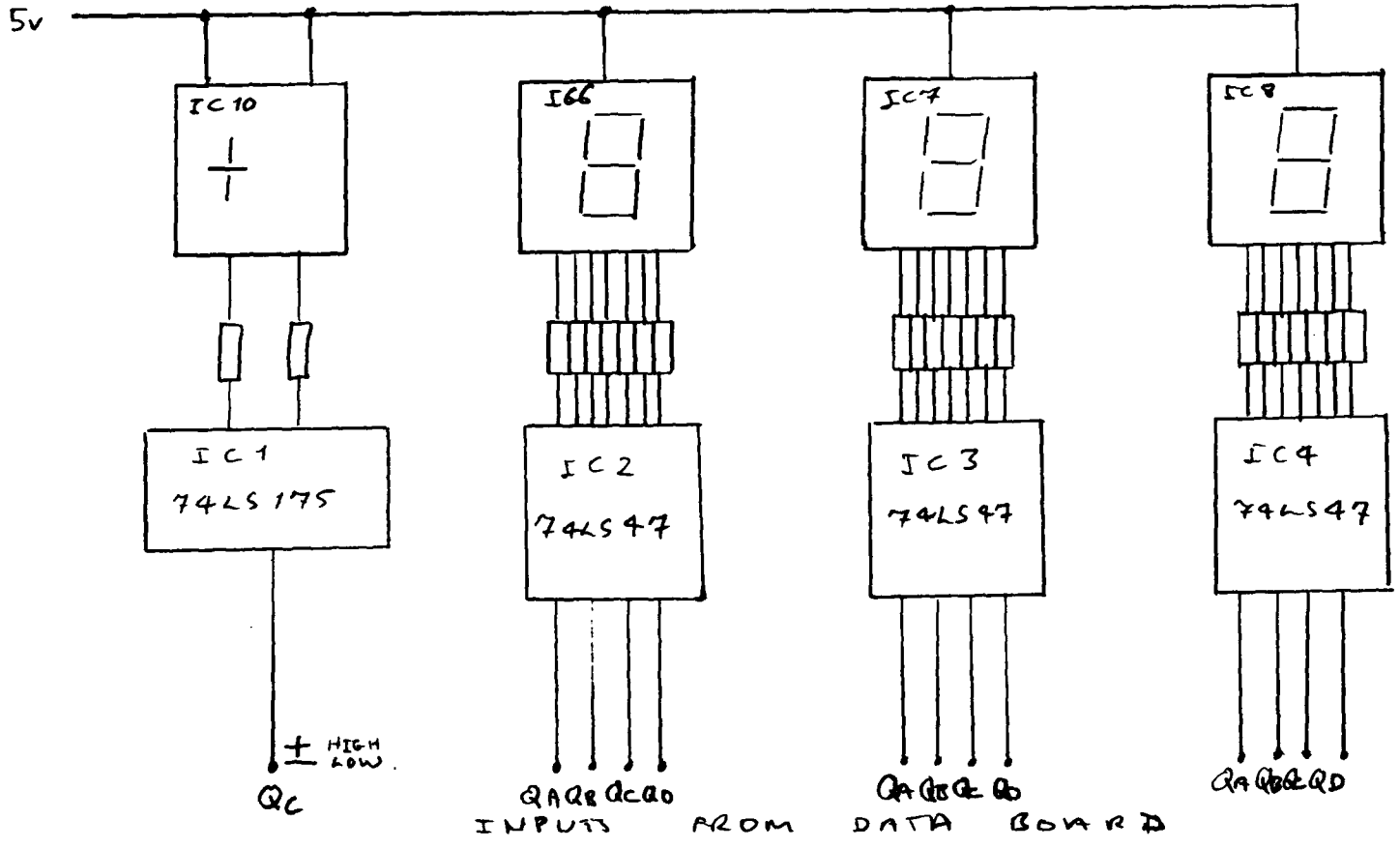


Fig 7 Display circuit

Fig 7

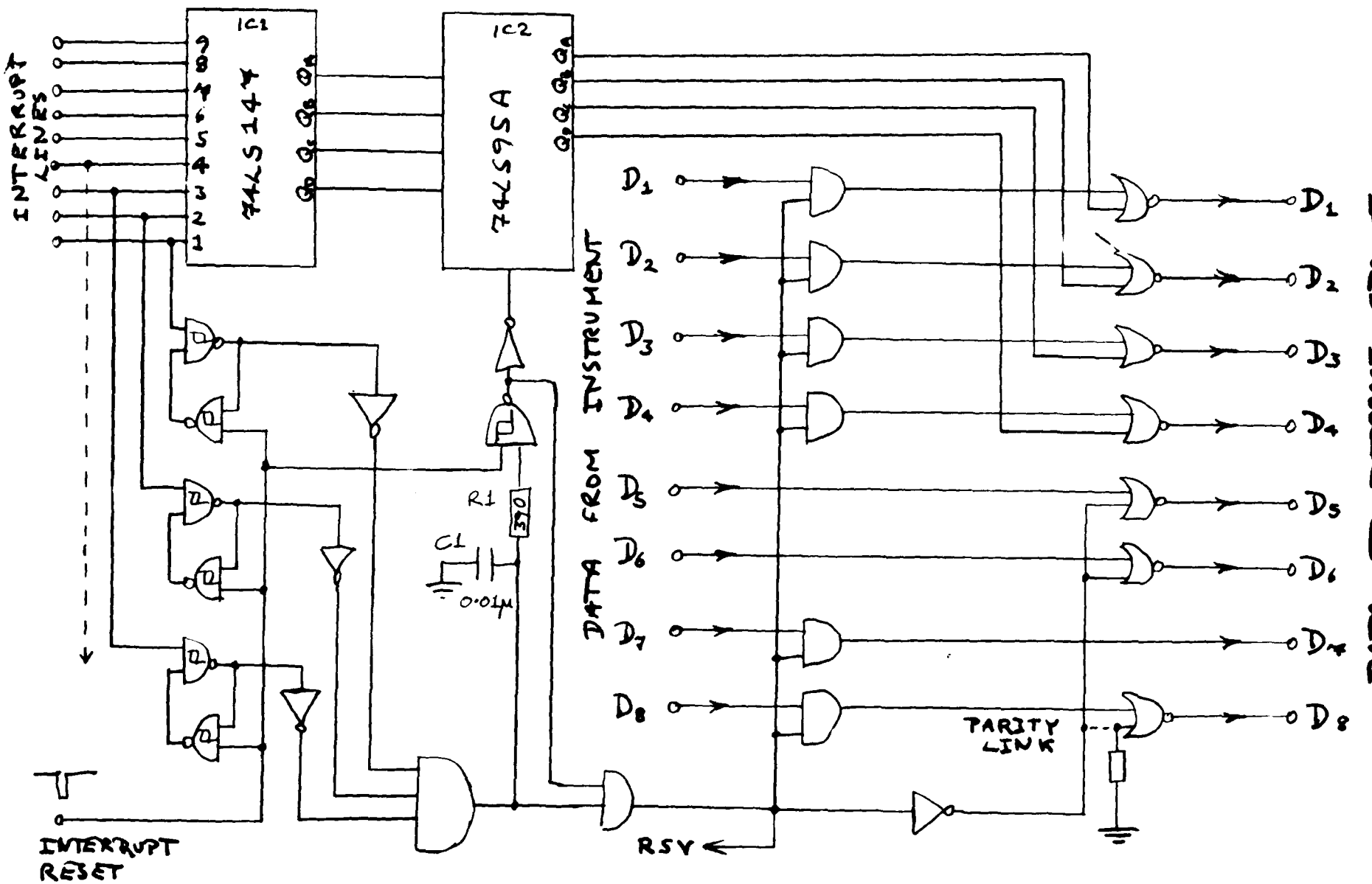


Fig 8 Interrupt encode circuit

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNLIMITED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TM Rad-Nav 168	N/A	UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aircraft Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	6a. Sponsoring Agency (Contract Authority) Name and Location
N/A	N/A

7. Title
A GPIB interface.

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Benson, A.M.			June 1981	24	3

Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		

15. Distribution statement
(a) Controlled by – Head of Radio & Navigation Department
(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)
Digital interface. GPIB. Rate table control.

17. Abstract

This Memorandum shows the use of the ~~Fairchild~~ 96LS488 integrated circuit in the design of a GPIB interface for use with a ~~Hewlett-Packard~~ System 35A desktop computer. Circuits are described which will convert GPIB data into parallel BCD data and provide facility for hardware interrupts. The interface will both transmit and receive data.

F5910/1

RAE Form A143


DA
FILM